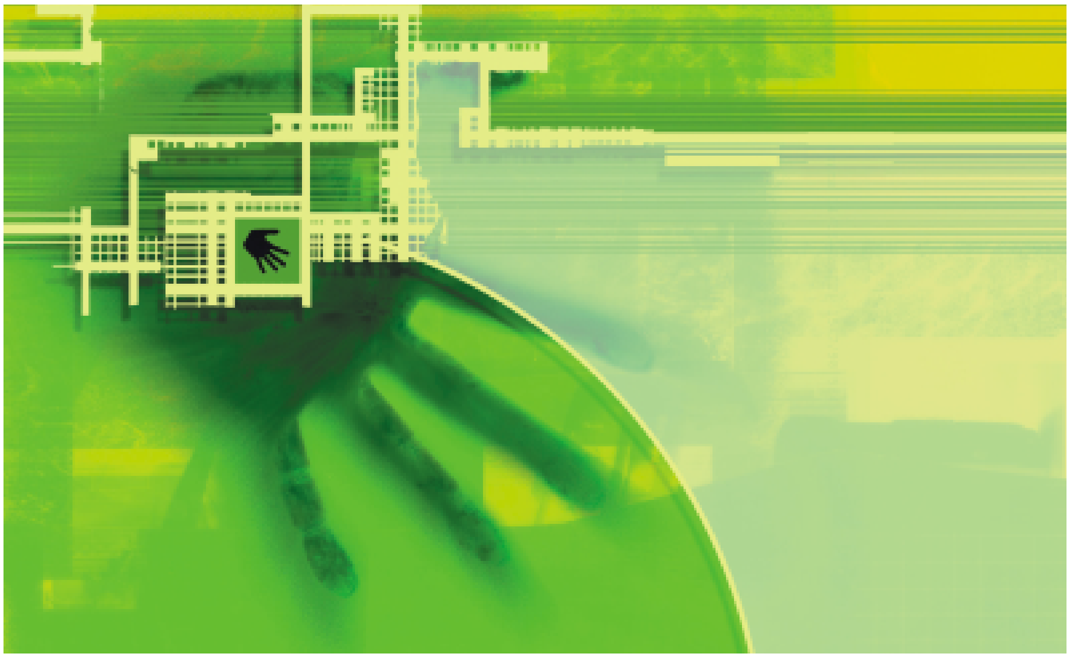
research alerts



Simon Osborne © Digital Vision Ltd.

The following abstracts are from recent issues and the forthcoming issue of ACM's *Transactions of Computer Human Interaction* (ToCHI). They are included here to alert *interactions*' readers to what research is being done in the field of Computer Human Interaction. The complete papers, when published, can be found in ACM's Digital Library at http://www.acm.org/pubs/contents/journals/tochi/1999-6/#3 and http://www.acm.org/pubs/citations/journals/tochi/1999-6-2/p133-dourish/.

Relative Contributions of Stereo, Lighting, and Background Scenes in Promoting 3-D Depth Visualization

The dramatic increase in the performance capabilities of contemporary computing hardware has enabled the development of various three-dimensional (3-D) user interface techniques for business, entertainment, and scientific applications[1] that were impossible just a few years ago. However, a significant challenge facing 3-D interface designers is to develop effective techniques to depict objects in 3-D space on a physical medium that is inherently two-dimensional (2-D): a flat computer screen. It is especially challenging to effectively depict 3-D depth relationships, such as relative position and size, among objects presented on a computer screen. To this end, interface designers use various secondary depth cues, including perspective, elevation, relative size, texture, shading and shadow, motion, and reference frames (see Kelsey [1993] for a thorough discussion of primary and secondary depth cues) to make 2-D objects appear 3-D.

We conducted two experiments to investigate the relative utility of moving objects that cast shadows on different background scenes so that the viewer could reposition and resize objects in space. For example, in Figure 1, the task was to reposition one of eight spheres in order to form a cube in a room background scene. Using a spaceball input device, human subjects "flew" the sphere located near the middle of the figure to form a cube by repositioning the misplaced sphere in the lower, rear, left-hand corner. The performance of each task was measured in accuracy and

[1] See http://nipper.gsfc.nasa.gov/vetdocs/VETHomePage.html and http://svs.gsfc.nasa.gov

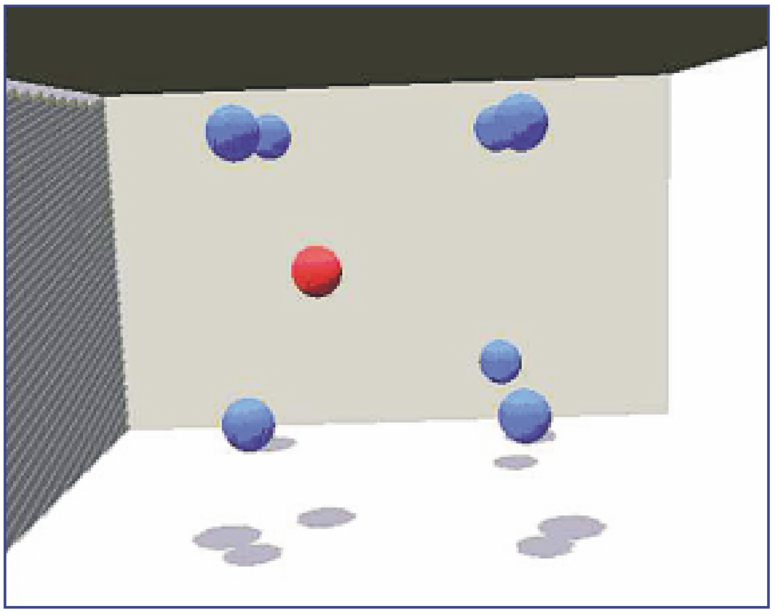


Figure 1. Repositioning one of eight spheres that cast shadows to form a cube with spheres at each cube vertex.

response time. Subjects repositioned and resized the spheres under four conditions: (1) objects casting shadows on and off, (2) shadows from one or two light sources (nested within the 'shadows-on' condition in (1)), (3) stereoscopic and monoscopic viewing, and (4) different background scenes (i.e. flat plane, room, stairsteps). The subjects had realistic stereoscopic viewing using a Silicon Graphics, Inc. Onyx2 workstation with Infinite Reality2 hardware, custom-developed OpenGL software, and 120-Hz, flicker-free, stereoscopic CrystalEyestm glasses manufactured by StereoGraphics.

The results of the two experiments have practical implications for designers of 3-D user interfaces. Whereas evidence existed that an object casting a single shadow can aid in positioning that object in space, evidence also existed that these performance benefits rapidly degrade as the number of shadows cast by that object, or as background scene complexity, increases. Additional findings of this research that are relevant to the design of 3-D user interfaces include the following:

- Adding one light source instead of no lighting improved the accuracy of object positioning regardless of background scene configuration.
- Adding a second light source never improved positioning performance, but rather, often impaired it.
- Introducing one light source to a monoscopically viewed background scene improved the accuracy of resizing to a level near that of stereoscopically viewed resizing.
- Adding a second light source rendered monoscopic resizing performance that was as poor as if there were no lighting.
- Stereoscopic viewing was a dominant depth cue, superior to monoscopic viewing, and to any shadow or background scene condition, for improving the accuracy of positioning and resizing and response time.

Geoffrey S. Hubona
Virginia Commonwealth University

Philip N. Wheeler
AppNet, Inc.

Gregory W. Shirah and Matthew Brandt
NASA–Goddard Space Flight Center

Reference

Kelsey, C.A. 1993. Detection of visual information. In *The Perception of Visual Information*, W.R. Hendee and P. Wells, eds. New York: Springer-Verlag, pp. 30–51.

Presto: An Experimental Architecture for Fluid, Interactive Document Spaces

If you ask someone what their favorite hobby or leisure activity is, it's pretty unlikely that they'd answer "filing." Filing is a chore, a necessary evil. Computers, of course, are meant to relieve us of these sorts of burdens, but the irony is that in fact, when I use the computer I end up doing a lot *more* filing, not less. Every time I store a piece of information—a document I've written, an e-mail message I've received, or the URL of an interesting Web page—I have to file it somewhere. I need to put it somewhere in the hierarchical structures that appear all over the place in today's systems—file system structures, e-mail folders, and bookmark stores. What's more, when I want to find it again, I need to remember just where I put it. Hmm, do I put papers about the Presto project in the **Presto** folder or the

papers folder? What about e-mail about the project?

Trees—hierarchies—are the sort of structure that only a computer scientist could love. They're quite efficient, of course, and back when computer systems were much less powerful than they are today, using trees was the only effective way to manage information. But in the era of desktop supercomputers, and with PCs getting faster all the time, it's long since time to find new ways to organize information that fit with how people want to organize it, not how the system wants to store it. The basic question that we have been addressing is, what happens when we eliminate these rigid structures? What are the alternatives, and what sorts of interaction do they support?

Of course, we all need structure as a way of managing information; as a way of understanding the relationship between data items. It's difficult enough to find our way around on the World Wide Web as it is; if we didn't have the structure imposed by Web pages, URL names, and hyperlinks, it would be totally impossible. The approach we have been exploring is using *properties,* tags that can be attached to documents (such as files and Web pages) stored in our system. Properties are things you already know about your documents: this one's important, that one's about Presto, another is an e-mail message or should be shared with Alice. Properties can be added directly by users, or automatically by systems (e.g., a mail message parser). A document can have many different properties at the same time, reflecting the different roles it plays for people and the different activities for which it is relevant. So, properties can be used to organize a document space flexibly according to the different tasks in which users engage.

We've built a prototype system, called Presto, based on these ideas. Presto is partly a system for end-user information management, like the file system, and partly an infrastructure for application development, like a database system. Our paper explores the requirements for property-based interaction and discusses some of the ways we attempted to integrate a property-based model with the conventional approach we need to support legacy applications such as present-day productivity tools. It also discusses some of the lessons we learned that we've been using to develop the second generation of our Placeless Documents system.

Paul Dourish,
W. Keith Edwards,
Anthony LaMarca, and
Michael Salisbury
Xerox Palo Alto
Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
Tel: +1 650 812 4421
dourish@parc.xerox.com